

46. (Amended-Clean Text) A method as claimed in claim 34 wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said first process will override the data and/or program modules of said second process.

47. (Amended-Clean Text) A method as claimed in claim 34 wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said second process will override the data and/or program modules of said first process.

51. (Amended-Clean Text) A method as claimed in claim 48 wherein a said process is subject to an evolutionary operation that allows the process to run in the second hardware component.

52. (Amended-Clean Text) A method as claimed in claim 48 wherein said second hardware component is a memory storage device.

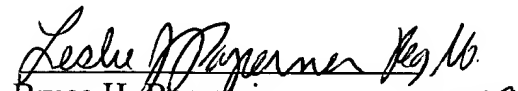
REMARKS

By the above amendment, the claims have been amended to delete multiple dependency.

P17529.A01

If there should be any questions, the Examiner is invited to contact the undersigned
at the telephone number listed below.

Respectfully submitted,
Hwee Hwa PANG and Teow Hin NGAIR


Bruce H. Bernstein
Reg. No. 29,027

June 11, 2001
GREENBLUM & BERNSTEIN, P.L.C.
1941 Roland Clarke Place
Reston, VA 20191
(703) 716-1191



7. (Amended) A computing environment as claimed in claim 1 [any preceding claim] wherein said evolutionary operations include the selective loading or reloading of objects into said process.

12. (Amended) A computing environment as claimed in claim 9 [any of claims 9 to 11] wherein said construct is provided with an authorising signature.

- 8 -

14. (Amended) A computing environment as claimed in claim 9 [any of claims 9 to 12] wherein after said construct is transferred the first process is suspended and the second process stored within said construct is be activated, and when the second process is concluded the data and program modules of the second process are added to the first process and the first process is re-activated.

15. (Amended) A computing environment as claimed in claim 9 [any of claims 9 to 12] wherein after said first process terminates at least some of the data and/or program modules from said first process are added to the second process stored in said construct and said second process is then activated.

19. (Amended) A computing environment as claimed in claim 16 [any of claims 16 to 18] wherein said data and said program modules from said second process are copied into said first process.

20. (Amended) A computing environment as claimed in claim 8 [any of claims 8 to 19] wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said first process will override the data and/or program modules of said second process.

21. (Amended) A computing environment as claimed in claim 8 [any of claims 8 to 19] wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said second process will override the data and/or program modules of said first process.

25. (Amended) A computing environment as claimed in claim 22 [any of claims 22 to 24] wherein a said process is subject to an evolutionary operation that allows the process to run in the second hardware component.

26. (Amended) A computing environment as claimed in claim 22 [any of claims 22 to 24] wherein said second hardware component is a memory storage device.

31. (Amended) A method as claimed in claim 28 [any of claims 28 to 30] wherein said construct is provided with an authorising signature.

32. (Amended) A method as claimed in claim 27 [any of claims 27 to 31] wherein a process is modified by the selective deletion of objects from within the process.

33. (Amended) A method as claimed in claim 27 [any of claims 27 to 31] wherein a process is modified by the selective loading or reloading of objects into a said process.

38. (Amended) A method as claimed in claim 35 [any of claims 35 to 37] wherein said construct is formed with an authorising signature.

39. (Amended) A method as claimed in claim 35 [any of claims 35 to 37] wherein after said construct is transferred the second process stored within said construct is caused to be activated within said first process.

40. (Amended) A method as claimed in claim 35 [any of claims 35 to 38] wherein after said construct is transferred the first process is suspended and the second process stored within said construct is activated, and when the second process is concluded the data and program modules of the second process are added to the first process and the first process is re-activated.

41. (Amended) A method as claimed in claim 35 [any of claims 35 to 38] wherein after said first process terminates at least some of the data and/or program modules from said first process are added to the second process stored in said construct and said second process is then activated.

45. (Amended) A method as claimed in claim 42 [any of claims 42 to 44] wherein said at least some data and/or program modules from said second process are copied into said first process.

46. (Amended) A method as claimed in claim 34 [any of claims 34 to 45] wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said first process will override the data and/or program modules of said second process.

47. (Amended) A method as claimed in claim 34 [any of claims 34 to 45] wherein in the event of a conflict between data and/or program modules of said first process and data and/or program modules of said second process, the data and/or program modules of said second process will override the data and/or program modules of said first process.

51. (Amended) A method as claimed in claim 48 [any of claims 48 to 50] wherein a said process is subject to an evolutionary operation that allows the process to run in the second hardware component.

52. (Amended) A method as claimed in claim 48 [any of claims 48 to 50] wherein said second hardware component is a memory storage device.